CLAIMS:

What is claimed is:

1. A method of improving performance in a Java computer program, comprising the steps of:

obtaining information associated with garbage collection; and

deducing changes in performance that will result from modifying the Java computer program.

2. The method of claim 1, wherein a cost of garbage collection to program performance is estimated using a duration of an average garbage collection event and a frequency of garbage collection events.

3. The method of claim 2, wherein the cost of garbage collection is reduced by reducing either or both of the duration and frequency.

4. The method of claim 2, wherein the duration depends on an amount of garbage that must be cleaned up, an algorithm used to do the collecting or copying, a heap compaction, a cost of reconciling object references that are moved, and a number of finalizers that must be executed.

5. The method of claim 2, wherein the frequency depends on the rate of object creation, the heap fragmentation, the size of the heap, and the garbage collection policy.

6. The method of claim 1, wherein the Java computer program is changed by reducing memory from a footprint of the Java computer program.

7. The method of claim 6, wherein given the amount of memory to be reduced from the footprint, a total duration for a run, and how much of the run is spent in garbage collection, the number of additional transactions that can be computed during the run is determined.

8. The method of claim 7, wherein the information associated with garbage collection is obtained from a verbosegc.

9. The method of claim 1, further comprising the step of:

modifying the Java computer program.

10. A computer system capable of running a Java program, comprising:

a garbage heap associated with garbage collection events, wherein garbage collection events have an average duration and frequency;

instructions for estimating changes in performance resulting from modification of the Java program using information obtained about the garbage collection events.

11. The system of claim 10, wherein a cost of garbage collection is reduced by reducing either or both of the duration and frequency.

12. The system of claim 11, wherein the duration depends on an amount of garbage that must be cleaned up, an algorithm used to do the collecting or copying, a heap compaction, a cost of reconciling object references that are moved, and a number of finalizers that must be executed.

13. The system of claim 11, wherein the frequency depends on the rate of object creation, the heap fragmentation, the size of the heap, and the garbage collection policy.

14. The method of claim 10, wherein the Java computer program is changed by deducting memory from a footprint of the Java computer program.

15. The method of claim 14, wherein given the amount of memory to be deducted from the footprint, a total duration for a run, and how much of the run is spent in garbage collection, the number of additional transactions that can be computed during the run is determined.

16. The method of claim 15, wherein the information associated with garbage collection is obtained from a verbosegc.

17. A computer program product in a computer readable medium for improving performance in a Java computer program, comprising the steps of:

first instructions for obtaining information associated with garbage collection;

second instructions for deducing changes in performance that will result from modifying the Java computer program;

wherein the Java computer program is changed by deducting memory from a footprint of the Java computer program.

18. The method of claim 17, wherein a cost of garbage collection to program performance is estimated using a duration of an average garbage collection event and a frequency of garbage collection events.

19. The method of claim 18, wherein the cost of garbage collection is reduced by reducing either or both of the duration and frequency.

20. The method of claim 18, wherein the duration depends on an amount of garbage that must be cleaned up, an algorithm used to do the collecting or copying, a heap compaction, a cost of reconciling object references that are moved, and a number of finalizers that must be executed.

21. The method of claim 18, wherein the frequency depends on the rate of object creation, the heap fragmentation, the size of the heap, and the garbage collection policy.

22. The method of claim 17, wherein given the amount of memory to be deducted from the footprint, a total duration for a run, and how much of the run is spent in garbage collection, the number of additional transactions that can be computed during the run is determined.

23. The method of claim 22, wherein the information associated with garbage collection is obtained from a verbosegc.